Comments on the security of RMAC as proposed in NIST Draft 800-38B

Éliane Jaulmes, Antoine Joux and Frédéric Valette

DCSSI Crypto Lab
18, rue du Dr. Zamenhof
F-92131 Issy-Les-Moulineaux.
email: eliane.jaulmes@wanadoo.fr
antoine.joux@m4x.org
fred.valette@wanadoo.fr

Abstract. In this brief note, we examine the RMAC draft standard as proposed in NIST Draft 800-38B. We review the draft according to the general properties of RMAC that we proposed in our paper at FSE 2002 (Full version on ePrint). In particular, we examine some important differences between the construction and the security claims.

1 Introduction

RMAC is a mode of operation that allows the construction of randomized message authentication codes from a good block cipher. It was originally proposed as a response to the call for modes of operation that was sent by the NIST in order to complement the **AES** effort.

We recall the description of RMAC and some notations. Let E be a block cipher with keys of size k and blocks of size b. E_K represents the block cipher used with the key K. Let $E_{(K,R)}$ represent the same block cipher with the key K and an additional parameter R of size r. It will later be explained how $E_{(K,R)}$ relates to E_K. Let $\mathcal{M} = (M_1, \ldots, M_l)$ be a padded message. The RMAC algorithm computes a message authentication code $\mathcal{C}$ as follows:

$$\begin{aligned} R &= \text{output of a random number generator}, \\ C_1 &= E_{K_1}(M_1), \\ C_i &= E_{K_1}(M_i \oplus C_{i-1}) \text{ for } i \text{ from 2 to } l, \\ m &= E_{(K_2,R)}(C_l), \\ \mathcal{C} &= (m, R). \end{aligned}$$

2 Revisiting RMAC with tweakable block ciphers

The security proof of RMAC [2] makes strong assumptions on the block cipher used in the construction. These assumptions can be summarized using the new notion of tweakable block cipher introduced by Liskov, Rivest and Wagner at Crypto 2002 [6]. Namely, the chosen block cipher using the salt R as a tweak, that is the block cipher used in the configuration of the last computation $E_{(K_2,R)}$, must be a good tweakable block cipher. In particular, the block cipher shall not suffer from any related key attacks.

In [2] two ways of computing $E_{(K_2,R)}$ are suggested for use with the **AES**.

First construction. The first suggestion is to use **AES** with keys of size 256 bits. In that case, K_2 and R may both be chosen of size 128 bits and we set $E_{(K_2,R)} = AES_{K_2||R}$. Another possibility would be to select a key K_2 of size 256 bits, an R of size 128 bits but padded with zeroes up to 256 bits and apply $E_{(K_2,R)} = AES_{K_2 \oplus R}$. From a security point of view, these two solutions are equivalent.

Second construction. The second suggestion is to use **AES** with keys of size 128 bits. In that case, K_2 and R are also of size 128 bits and we set $E_{(K_2,R)} = AES_{K_2 \oplus R}$.

For the first construction, the security proof is given in the standard model. For the second construction, the security proof is done under a much stronger assumption in the ideal cipher model. Thus, while more costly in terms of computations, the first construction has to be preferred when possible.

There is no construction proposed in [2] for use with the DES or Triple-DES. In fact, it is unclear whether tweakable block ciphers can be formed from these algorithms. While we understand the need of allowing DES or Triple-DES for backward compatibility concerns, it should be made clear in the final draft that this choice greatly damages the security of the construction.

3 Security level of the construction

The first construction is proven secure in the standard model. The second construction is proven secure in the ideal cipher model. Both security proofs states that a computational power at least approximately equal to $2^n/n$, where $n = \min(k, b)$, is needed to break the scheme. It should be noted that increasing the size of the key in the **AES** beyond 128 bits will not provide an increase of security.

In [8, Appendix A], it is stated that an exhaustive key search on RMAC would require 2^{2k-1} computations. While this is true of a brute force search with no optimization, it is misleading since other attacks exist that require much less computation power. This remark also stands for the forgery claims. In fact, we believe that forgery and key recovery can both be achieved with high probability in approximately $2^n/n$ computations.

4 MAC truncation

It is possible to reduce the size of the MAC by using some truncation on R and m. The idea is to reduce the size of the MAC while minimizing the security loss.

In order to do this, we need to consider not only collision based attacks but also plain guessing attacks. Collision based attacks work by colliding both R and the output of the last AES evaluation, while guessing attacks need to correctly guess an m-bit value. In order to be as secure as possible, we balance the probability of success of the two kinds of attacks. The size of the blocks is n. Let r be the number of bits of the random R and let s be the number of bits of the output m that are kept to form the MAC. We want $(r+n)/2 = s$. For a security of $2^{-(n-d)}$ we take $r = n - 2d$ and $s = n - d$ and that gives us a MAC size of $t = r + s = 2n - 3d$.

For example, if $d = 43$, we get a security of 2^{-85} with a MAC tag of size $t = 127$. If we add one bit to avoid padding messages of length multiple of 128, as explained in [2], we get a MAC tag of size 128 bits.

Of course, the above evaluation of the security level for the truncations is a crude approximation and should not be taken as a security proof. In particular, it should be noted that some factor, depending on r and on the number MAC computations that the adversary can do, should apply and reduce the security.

5 Message padding

It is possible to avoid the padding of already complete message strings by using the technique proposed in [2], consisting in adding one bit to R depending on whether or not the message had to be padded. This technique could be described and proposed in the NIST Draft in order to obtain a better efficiency for RMAC. It would address one of the concerns regarding efficiency expressed in [9] by Rogaway.

6 Addressing remarks raised as public comments

Four public comments [7,5,4,9] where posted regarding the NIST Draft 800-38B.

Lloyd describes in [7] a forgery attack and a key recovery attack against RMAC of complexity approximately 2^k. These attacks are completely compatible with the bound of the security proof that is equal to $2^n/n$ where $n = \min(b, k)$.

Knudsen [4] pointed out security problems arising from the choice of Triple-DES. He also presents an attacks requiring $2^n/n$ computations. This attacks fits the security bound of the proof.

Kohno describes in [5] key-collisions attacks that are a variant of Biham's attack [1] and also fits within the bound of the security proof.

In [9], Rogaway suggests to abandon the RMAC proposal. He describes the NIST RMAC proposal as a *salted* RMAC. In his security definition, he gives the adversary control on the salt R. It is true that the proof of [2] does not apply in case R is controlled by the adversary. However we believe that only the two following properties are important for the value R. Namely that, first, a given R should not be repeated more than n times during 2^n MAC computations and that, second, the adversary may not be allowed to choose the value of R. While we have no proof that RMAC is indeed a *salted* MAC, according to his definition, we believe that the first construction (in the standard model) from section 2 is in fact secure in this model.

Rogaway claims that the ideal cipher model is not suitable for a MAC proof of security. Is is true that the proof in the ideal cipher model makes some strong assumptions on the block cipher **AES**. However we believe that those assumptions are what should be expected of a good block cipher and that the **AES** is likely to meet these requirements.

References

1. E. Biham. How to decrypt or even substitute DES-encrypted messages in 2^{28} steps. *Information Processing Letters*, 84, 2002.
2. É. Jaulmes, A. Joux, and F. Valette. On the security of randomized CBC-MAC beyond the birthday paradox limit – A new construction. Cryptology ePrint Archive, Report 2001/074, revised 28 Nov 2002. Full version of [3].
3. Éliane Jaulmes, Antoine Joux, and Frédéric Valette. On the security of randomized CBC-MAC beyond the birthday paradox limit – A new construction. In *Fast Software Encryption'02*, Lecture Notes in Computer Science, 2002.
4. L.R. Knudsen. Analysis of RMAC. NIST Modes of Operation, Comment on Draft SP 800-38B, November 2002.
5. T. Kohno. Key-collision attacks against RMAC. Cryptology ePrint Archive, Report 2002/159, 2002.
6. M. Liskov, R. Rivest, and D.Wagner. Tweakable block ciphers. In Moti Yung, editor, *Advances in Cryptology — Proceedings of CRYPTO 2002*, volume 2442 of *Lecture Notes in Computer Science*, pages 31–46. Springer, 2002.
7. J. Lloyd. An analysis of RMAC. Cryptology ePrint Archive, Report 2002/170, November 2002.
8. NIST Special Publication 800-38B, Springfield, Virginia. *NIST. DRAFT Recommendation for Block-Cipher Modes of Operation: The RMAC Authentication Mode*, October 2002.
9. P. Rogaway. Comments on NIST's RMAC proposal. NIST Modes of Operation, Comment on Draft SP 800-38B, November 2002.